UTILITY APPLICATION

OF

MIKE JADON

ROBERT LERCARI

RICHARD M. MATHEWS

WILLIAM R. PEEBLES

AND

PHAP NGUYEN

FOR

UNITED STATES PATENT

ON

# DISK-ARRAY CONTROLLER WITH HOST-CONTROLLED NVRAM

Sheets of Drawings: 6

SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
333 South Hope Street, 48th Floor
Los Angeles, California 90071
(213) 620-1780

# DISK-ARRAY CONTROLLER WITH HOST-CONTROLLED NVRAM

## Cross-Reference to Related Application

**[0001]** Priority is claimed under 35 U.S.C. § 119(e) to United States Provisional Patent Application No. 60/494,696, filed on August 13, 2003, entitled "Memory Card and Related Methods for Using It" by Mike Jadon, which is incorporated by reference herein.

## Technical Field of the Invention

**[0002]** The present invention relates generally to peripheral controllers for data storage. More particularly, it relates to enhancing synchronous I/O operations to disk-array controllers.

## Background of the Invention

**[0003]** There is very great demand for high-speed stable storage. Disks provide stable storage, but latency and transfer times can be high.

**[0004]** Non-volatile random-access memory (NVRAM) can be use to improve performance in a number of ways to improve response time and data reliability in server appliances. NVRAM may consist of random-access memory that does not require power to retain data or Dynamic Random-Access Memory (DRAM) or Synchronous DRAM (SDRAM) that has secondary power such as battery or an external universal power supply (UPS).

**[0005]** One such prior-art application is shown in Figure 1. The host computer 11 may write important data to disks 17. When time is critical, it may instead store data to the faster NVRAM device 12. The DMA memory controller 18 manages the NVRAM 19 and provides direct memory access (DMA) services. DMA is used to transfer data in either direction between host memory 15 and NVRAM 19 across an industry-standard peripheral component interconnect (PCI) bus 13. DMA performs transfers while the host computer 11 performs other operations, relieving the host computer 11 of those duties. The data stored in NVRAM 19 may be a cache of

data that will eventually be written to disks 17, a journal of changes to the disks 19 that may be replayed to recover from a system failure but which never needs to be written to disks 17, or other information about transactions that may eventually be processed causing related data to be written to disks 17.

**[0006]** This application allows the host computer 11 to directly control the NVRAM device 12, but it does not allow the NVRAM 19 to be used together efficiently with the disks 17. Data moving from NVRAM to disk must pass through the primary bus 13. This can reduce performance because the bus must be shared with other device transactions. Another disadvantage of this scheme is that NVRAM device 12 requires its own location on the primary bus 13 rather than sharing one with the controller for the disks 17. Locations on the bus often are not easily made available.

**[0007]** Figure 2A shows a prior-art implementation in which NVRAM is attached to a storage device. The host computer 100 is attached to a disk controller 101 by an interface 104, possibly a PCI bus. The disk controller is attached to a disk or other storage device 102. The interface 105 may be a local bus such as Small Computer System Interface (SCSI) or AT-attached (ATA). The disk 102 may also be replaced by an intelligent storage device such as network-attached storage (NAS) or a storage area network (SAN) device. In this case interface 105 may be a network or fibre channel connection. The NVRAM 103 is under complete control of the disk or storage device 102. The host computer 100 has no way to access the NVRAM contents using interface 105.

**[0008]** Figure 2B is similar to Figure 2A except that the NVRAM 203 has moved to the disk controller 201. The disk controller may manage disks 202 as a JBOD (Just a Bunch of Disks) or a RAID (Redundant Array of Independent Disks) system. When the host computer 200 makes a request to the disk controller 201, the controller may choose to cache data in the NVRAM 203. Management of the NVRAM is the responsibility of the disk controller. This includes algorithms for deciding when data cached in NVRAM will be transferred to disk and when it will be discarded.

[0009] The solutions in Figures 2A and 2B solve the problem of keeping the NVRAM data close to the disks, but they take control of the NVRAM away from the host computer. Usually the host computer has a much better idea of how data is being used than does the disk or the disk controller. The host can know if data is temporary in nature and never needs to be copied to disk. The host can know if the data is likely to be modified again soon and thus disk accesses can be reduced if the data is not immediately copied to disk. The host can know if data will no longer be needed and can be removed from cache once it is on disk.

[0010] There are other prior art applications that utilize bus bridges. These bus bridges often include local memory that is a subset of the bridge. Figure 3 illustrates a host computer 250 that connects to one or more devices 252 through a PCI bus bridge. Information on PCI bus 254 is forwarded by the bridge 251 to PCI bus 255 as necessary to reach the target device 252. Information on PCI bus 255 is forwarded by the bridge 251 to PCI bus 254 as necessary to reach the host computer 250. The PCI bridge 251 may use local bridge memory 253 temporarily to store the data that flows through the bridge. Data coming from bus 254, for example, may be stored in the bridge's memory until bus 255 is available and device 252 is ready to receive the data. This memory is used by the PCI bridge 251 to make its routing function more efficient. There is no way for the host computer 250 to directly control this memory, specifically where the bridge 251 puts this data or when it is removed from memory 253. From the perspective of the host computer 250, it is writing the data directly to the device 252 except for a time delay in having the data reach the device. While the present invention utilizes some of these same bus bridge devices with associated local memory, it should be noted that the local bus bridge memory 253 is a subset of the bridge that is transparent to the host computer. This is unlike NVRAM 19 in Figure 1 or NVRAM 309 in Figure 4, which are endpoint devices that can be directly controlled by the host computer.

[0011] Accordingly, it is an object of the present invention to provide NVRAM that may be fully controlled by the host computer.

[0012] Another object of the present invention is to provide NVRAM and disk controllers connected by private data paths while allowing each to run on its bus at as high a speed as possible.

[0013] Another object of the present invention is to provide NVRAM and disk controllers that may share a single connection to the host computer's primary bus.

Summary of the Invention

[0014] The present invention combines NVRAM under control of the host computer with disk array controllers close to the NVRAM. Unlike many disk/RAID controllers that have a processor that takes control of the NVRAM, the present invention leaves the NVRAM to be used by the host. A plurality of private buses is used in the present invention to allow the host computer to program the NVRAM and disk array controllers to transfer data directly between themselves. Either the disk array controllers or the NVRAM controller may act as DMA masters.

Brief Description of the Drawings

[0015] Figure 1 is a block diagram of a prior art PCI NVRAM device.

[0016] Figure 2A illustrates a prior art disk device or storage device that includes NVRAM.

[0017] Figure 2B illustrates a prior art disk controller or RAID controller that includes NVRAM.

[0018] Figure 3 illustrates a prior art PCI bridge with SDRAM.

[0019] Figure 4 is a block diagram of a preferred embodiment of the invention.

[0020] Figure 5 is a flow chart for allocating NVRAM.

[0021] Figure 6 is a flow chart for scheduling writes to disk.

[0022] Figure 7 is a flow chart for choosing whether to keep data in NVRAM.

Detailed Description of the Invention

**[0023]** Figure 4 illustrates a preferred embodiment of the invention incorporated into a Server System 300. The Host Computer 301 includes a Primary PCI Bus 303, though other bus technologies may be used. Attached to the bus 303 is the Host-NVRAM Disk-Array Controller 302. Within this controller 302 are a plurality of local PCI buses 307, though again it is understood that other bus technologies may be used.

**[0024]** A plurality of PCI bridges 304A, 304B, through 304N connects the various buses. The bridges are used to meet load requirements on each bus that limit the number of devices that may be attached to the bus. The bridges also may be used to connect buses of different technologies or different speeds. For example, some devices on the controller 302 may use the PCI 2.2 specification while others use the PCI-X 2.0 specification.

**[0025]** A plurality of disk-array controllers 310A, 310B, through 310N are attached to the plurality of PCI buses 307. In the preferred embodiment these are SCSI controllers or multi-port Serial ATA (SATA) controllers.

**[0026]** The DMA memory controller 308 manages the NVRAM 309. The NVRAM may consist of memory that requires no power to maintain data (such as magnetic memory), battery-backed SDRAM, or other RAM that uses external power. The preferred embodiment shown uses either power from the host computer 301 or rechargeable batteries 312, with a power regulator 311 managing the delivery of power to the NVRAM and to the battery recharge circuit.

**[0027]** The memory controller 308 includes DMA master capabilities that allow direct memory transfers between NVRAM 309 and host memory 315 or between NVRAM 309 and the plurality of disk array controllers 310 via one or more of the plurality of buses 307. The host computer 301 controls the NVRAM 309 and may program the DMA memory controller 308.

[0028] The NVRAM 309 may also be accessed as a target by either the host computer 301 or the disk array controllers 310. This allows NVRAM to be used as ordinary memory. Unlike cache on a disk or disk controller, this allows it to be accessed one byte at a time rather than in large blocks. The entire NVRAM may be mapped into the address space of the bus, though in the preferred embodiment only a window into NVRAM is mapped. A register in the NVRAM controller 308 determines which window is visible.

[0029] The host can use any method for caching data to NVRAM. The advantage of the present invention is being able to keep the host's cache close to the disk controllers. Because the host controls the cache, it can determine what data is to be cached, when the cached data is to move to or from the disk controllers, and when it can be freed.

[0030] Because the NVRAM cache appears to the host as ordinary memory, the host can access individual bytes of data in the cache. On the other hand, prior art disk-based cache must generally be accessed in blocks of 512 bytes or larger.

[0031] Figures 5 through 7 illustrate typical algorithms that may be used to manage a cache. The advantage of the present invention is that the host computer 301 is able to make these decisions rather than a disk or disk controller. In a preferred embodiment of the invention, the host computer 301 would allocate memory from the NVRAM 309 (Figure 5). On boot of the host computer, it would recognize data already allocated in the NVRAM. Data that needs to be stored quickly can be written to NVRAM. In some cases, such as data from a file system journal, the data may be expected to become obsolete in a short time, as determined in step 501. In such a case, the host computer 301 may never send the data to disks. The host may schedule other data that needs to be kept for a long time to be transferred to disk. If the host does not expect the data to be used again or modified again soon, the host may choose to do the transfer immediately, as in step 502. For other types of data, the host may choose to delay the transfer to disk, as in step 503. When writes are delayed, it may be desirable not to do the write until it is

necessary to free space in step 401. Once the data is on disk, if it is determined in step 601 that the data is not likely to be needed again soon, the NVRAM can be freed; otherwise, the NVRAM may remain available for the host to read the data faster than from disk. When data needs to be read from storage system, the host will also recognize when copies of the data are still in NVRAM and thus the data can be retrieved more quickly than going to disk.

**[0032]** The preferred embodiment will include storing file system journals to NVRAM that are never transferred to disk. It will include storing file system changes in NVRAM in which the same data is modified frequently such as access time on files or changes associated with creating or deleting large numbers of files. The host computer will send these changes to disk less frequently, but the changes will be preserved in stable storage in the NVRAM. The preferred embodiment will include saving transactions in NVRAM even before processing is complete on incorporating the transactions into complex databases or other files. It will also include using NVRAM to create a checkpoint of data on disk, with all updates going only to NVRAM while the disk contents are copied such as when creating a backup.

**[0033]** The methods above are not by themselves new. The advantage of the present invention is that the host computer 301 is better able to make each of the decisions involved than the disk or disk controller. The host retains control of these decisions while having the convenience of having the data stored close to the disk controllers. Applying these methods to a host-controlled cache rather than a disk-controlled cache provides advantages in performance.